

UTILITY PATENT APPLICATION of

Li Gong

For

TITLE: System and method for Concurrent Version Control and Information Management of files and documents sent as attachments through e-mail or web-mail

CROSS-REFERENCE TO RELATED APPLICATIONS: Provisional patent application No. 60/393,546, confirmation No. 8023

FEDERALLY SPONSORED RESEARCH: Not Applicable

SEQUENCE LISTING OR PROGRAM: Not Applicable

BACKGROUND:

Various problems can arise when using e-mail attachments to send out information that needs constant update and modification.

One of the problems is lacking effective direct document management (version control. Edit, Delete, and etc.). Assume one user sends an original document as attachment to another user. The recipient then downloads the document, modifies it and sends the modified version back to the original sender. The original sender gets and modifies the document, and sends it back to the same recipient. This process may take many rounds. Soon, those two users may have sent the same file but with modifications to each other many times. Up to this point, each user has several versions of the same document as e-mail attachments. It is not only very confusing as to how to manage the modified attachments, but also causing potential document integrity and authenticity concerns among legitimate senders and receivers.

Another potential problem is e-mail storage issue. Each e-mail user gets a limited amount of storage space for storing e-mails, especially when using web-based e-mail

applications, such as Yahoo mail or Hotmail. Sending one large document back and forth as attachment several times will quickly saturate allocated storage space. And users have to constantly clean up the mailbox by deleting entire e-mails. Other useful information can be lost during such a brutal clean-ups. An application is needed to allow user selectively delete attachments and leave other part of a email untouched.

Furthermore, attachments are referred to or related to some particular project(s). Without effectively aggregating and managing the attachment information from both project and participant perspectives and link attachments directly to project(s), lots of useful information may be lost.

Based on the foregoing, there is clearly a need to allow users directly apply file management controls (Concurrent Version Control or other file management rules/commands, such as Edit, Delete, Link, Associate, etc.) on attachments in a manner that avoids said file and storage space management problems inherent in use of e-mail attachments.

SUMMARY:

Using an e-mail or web-mail client, a user can send out messages with attachment(s) in any formats. Instead of sending the attachment together with the mail message through conventional mail servers, through user specification or pre-defined properties, the attachment will be detached from the original mail message content and be stored into a local storage area of a special Project/Information Management Server (**IMS**) that can be installed inside or outside companies' firewalls. A unique attachment descriptor and locator will be generated to identify the saved attachment, and the original e-mail will include the newly created descriptor and locator in its message body and be sent to recipients through normal mail servers. The Information Management Server will apply Concurrent Version Control (**CVS**) and other file and document management rules/commands (such as Edit, Delete, Modify, Link, Associate, etc.) specified by the e-mail sender on the stored attachment files. Upon receiving the e-mail, the recipients can click the attachment locator link and obtained a version-controlled copy of the original

attachment from the IMS. Recipients can modify the attachment in his/her local machine and check in the modified version as an attachment through e-mail or through provided Client Information Management Web Interface. All users having rights to access the attachments will receive e-mail notifications for any version or content update of a file. Permission to access the Client Information Management Web Interface will be administrated by the original e-mail creator or the group project organizer. IMS will manage and log all check-in, checkout and modification activities related to the attachment, and maintain one updated master copy. All previous versions will be compressed and stored until deletion. A user can admin all managed attachments through either e-mail or web-mail client, or Client Information Management Web Interface (login authentication is needed). Attachments can be managed under a specified project environment preemptively created by its organizer(s). A dynamic link of all projects (attachment related) to which a user subscribed will be conveniently displayed on e-mail or web-mail client interface as folders in addition to the Client Information Management Web Interface. The system allows direct e-mail attachment management.

DRAWINGS:

Figure 1: System and Flow chart of conventional E-mail system

Figure 2: System and Flow Chart of the claimed new system on how to apply file management controls on e-mail attachments

DETAILED DESCRIPTION:

FIELD OF THE INVENTION

The present invention relates to techniques for directly managing e-mail attachments by applying Concurrent Version Control and other file management rules (such as Edit, Delete, Modify, Link, Associate, etc.) using either E-mail or Web-mail client interface, or web-based information management interface.

BACKGROUND OF THE INVENTION

Various problems can arise when using e-mail attachments to send out information that needs continuous update and modification.

One of the problems is lacking effective direct document management (version control. Edit, Delete, and etc.). Assume one user sends an original document as attachment to another user. The recipient then downloads the document, modifies it and sends the modified version back to the original sender. The original sender gets and modifies the document, and sends it back to the same recipient. This process may take many rounds. Soon, those two users may have sent the same file but with modifications to each other many times. Up to this point, each user has several versions of the same document as e-mail attachments. It is not only very confusing as to how to manage the modified attachments, but also causing potential document integrity and authenticity concerns among legitimate senders and receivers.

Another potential problem is e-mail storage issue. Each e-mail user gets a limited amount of storage space for storing e-mails, especially when using web-based e-mail applications, such as Yahoo mail or Hotmail. Sending one large document back and forth as attachment several times will quickly saturate allocated storage space. And users have to constantly clean up the mailbox by deleting entire e-mails. Other useful information can be lost during such a brutal clean-ups. An application is needed to allow user selectively delete attachments and leave other part of a email untouched.

Furthermore, attachments are referred to or related to some particular project(s). Without effectively aggregating and managing the attachment information from both project and participant perspectives and link attachments directly to project(s), lots of useful information may be lost.

Based on the foregoing, there is clearly a need to allow users directly apply file management controls (Concurrent Version Control or other file management rules/commands) on attachments in a manner that avoids said file and storage space management problems inherent in use of e-mail attachments.

In order to demonstrate the benefits of the new system and method, it is necessary to show the difference between conventional e-mail system and the new system, and find out what are some of the difference by directly managing attachments. Figure 1 is the system and flow chart of conventional e-mail system. Figure 2 is the claimed new system and method for managing e-mail attachments directly.

Conventional E-mail System process (see Figure 1):

- 1, Mail client sends a message with attachment to a recipient.
- 2, Recipient Mail Server receives the message and store the attachment in recipient mail storage space.
- 3, Recipient gets the message and sends request to download the attachment.
- 4, the Mail server loads the attachment from the storage and passes it back to recipient's local machine.

There are three problems:

- If recipient modifies the attachment and sends back to the sender and so forth, it is difficult to make the modification synchronized, especially when more than two people involved
- Many mail applications leave copies of mails on server. So every time a modified attachment created, a copy will be stored on server, and pretty soon, it will eat up all storage space on the mail server.
- Users have to delete an entire e-mail in order to clean up the email storage space.

How the claimed new system and method for managing attachments work (See Figure 2):

1, user sends e-mails with attachments through e-mail client interface (Outlook, etc.) or web browser based web-mail client interface (Hotmail, etc.). There are two ways for users to specify any special file treatment rules on attached documents: by selecting from rule control feature list (this list may include features such as Concurrent Version Control etc.) that is directly linkable to the attached file names; or by sending or replying messages under special project folders that are created automatically each time user creates a new project.

1', Users can access the Client Project/Information Management Web Interface to manage attachment information (login authentication is needed).

2, The Adapter/Server Engine will process the e-mail and check whether the attachment is subjected to Concurrent Version Control or other specific rules or not: if yes, the attachment will be separated from the message body and a unique attachment locator and

descriptor will be generated, passed along with the original message and be processed by the Mail Server; if not, all information is passed and processed by the Mail Server as in ordinary situations.

3, Adapter/Server Engine passes the attachment files or documents along with identification information (descriptor and locator) to Project/Information Management Server.

4, The Project/Information Management Server will apply concurrent version control and other document management methods/rules, such as Edit, Delete, Modify, Link etc., to the document and store it in a structured environment. Users can access and change the environment setting by login through Client Information Management Web Interface.

5, Mail Server sends the message to recipients' Adapter Engine. The Adapter Engine will judge whether the attachment is a virtual link (descriptor and locator information only) or not. If it is not subjected to any version control or file management commands/rules, the Adapter Engine will simply serve as a mail proxy serve and relay the message to the recipient; if it is subjected to some kind of version control or file management commands/rules, the Adapter Engine will help retrieve and manage information (file or document) per user request;

6, Recipient gets the message and requires downloading the attached file(s).

6', Recipient(s) can access the attachment(s) directly through Client information Management Web Interface (login authentication needed). Under each client domain, as long as the owner of the original document (or project) permits modification by that client, all information related to that document (or project) is manageable and changeable.

7, Adapter/Server Engine will handle the download request and get a controlled version of the master attachment files from the Project/Information Management Server.

8, The Project/Information Management Server will pass the version controlled file(s) back to Adapter Engine, then to recipient local machine. The recipient can modify the file(s) in his/her local machine and check in the modified version through e-mail or through Client Information Management Web Interface. The Information Management server will store one master copy of the original file(s), and all modification will be checked against the master version and get updated. This ensures data integrity and less file storage space needed. It also opens up ways to manage e-mail attachment information directly into a leverage-able and share-able environment.